Projekt parkirnih senzora

SPECIFIKACIJA KODA

crtanje_quickstart.cpp

Borna Sirovec 2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

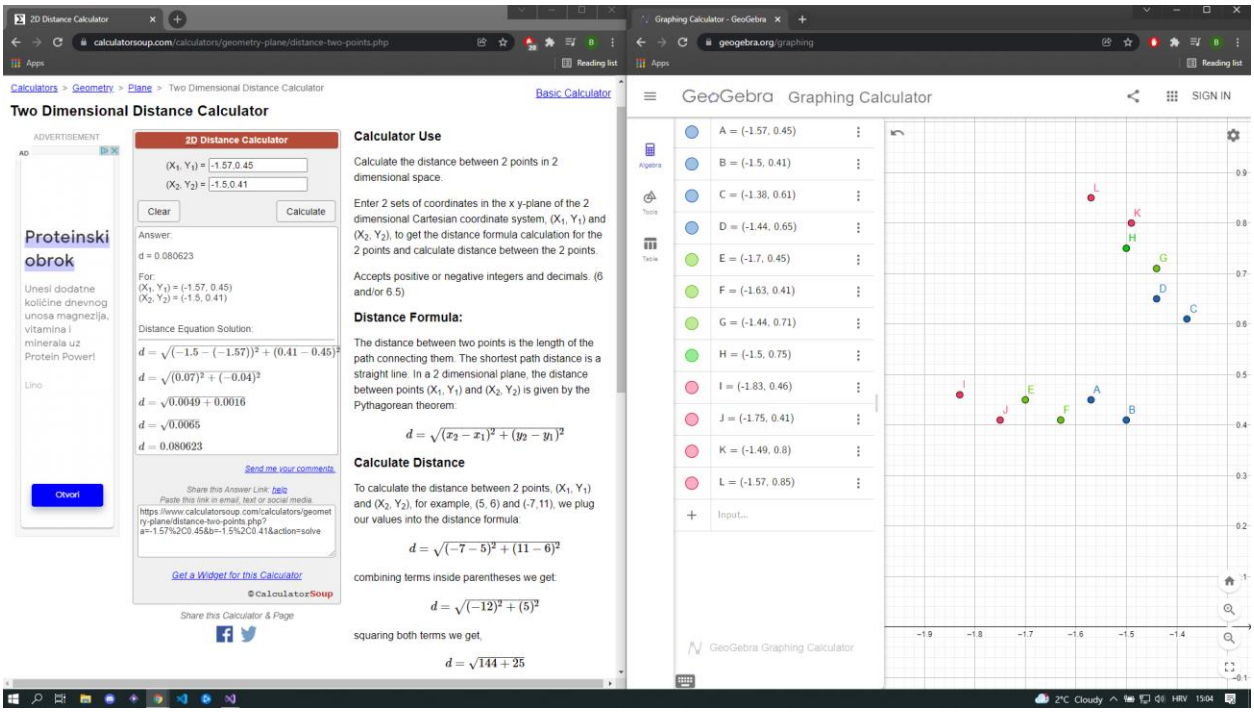
SADRŽAJ

UVOD

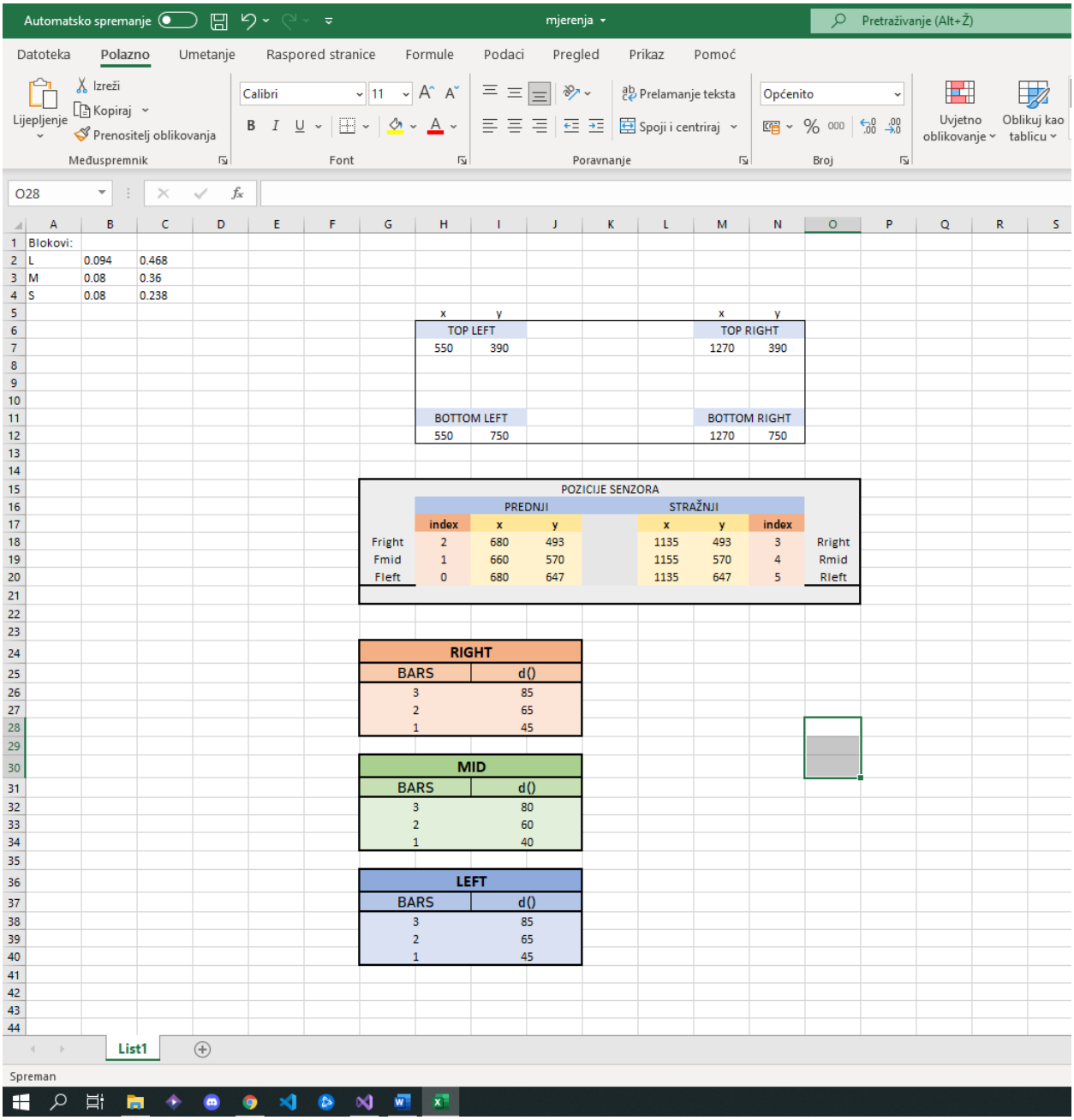
U ovoj ću specifikaciji ukratko prokomentirati kod, koje sam tehnike i funkcije koristio kako bi napravio simulaciju parking senzora. Budući da nisam pisao kod vezan za postavljanje slike i početnih okvira, to neću komentirati. Simulacija radi na način da se pokazivačem miša simuliraju prepreke te se s obzirom na udaljenost pojedinih senzora, na sučelju crtaju pravokutnici različitih boja i veličina, te se reproduciraju zvukovi različitih frekvencija.

Kako sam pojedine pravokutnike morao sam računati, pomoću alata GeoGebra računao sam koordinate točaka te udaljenosti između pojedinih pravokutnika. Te podatke, i podatke vezane za položaje senzora te udaljenosti na kojima se pale senzori spremao sam u excel tablicu.

Zvukovi su kreirani u virtualnoj mašini te su u progaramu Audacity podešeni (duljine izvedbe, lijeva ili desna slušalica)



Slika 1. Visualizacija pravokutnika u GeoGebri



Blokovi:		
L	0.094	0.468
M	0.08	0.36
S	0.08	0.238

	x	y
TOP LEFT	550	390
BOTTOM LEFT	550	750

	x	y
TOP RIGHT	1270	390
BOTTOM RIGHT	1270	750

POZICIJE SENZORA							
	PREDNJI			STRAŽNJI			
	index	x	y	x	y	index	
Fright	2	680	493	1135	493	3	Rright
Fmid	1	660	570	1155	570	4	Rmid
Fleft	0	680	647	1135	647	5	Rleft

RIGHT	
BARS	d()
3	85
2	65
1	45

MID	
BARS	d()
3	80
2	60
1	40

LEFT	
BARS	d()
3	85
2	65
1	45

Slika 2. Excel Tablica s podacima

1.Inicijalizacija

Simulacija radi na način da se konstantno računaju udaljenosti između položaja miša i senzora te se one spremaju u polje s udaljenostima. U ovisnosti o tim udaljenostima, dalje popunjavamo polje stanja senzora. Svaki element polja stanja predstavlja jedan senzor, te može poprimiti vrijednosti 0-3 odnosno koliko nam je miš blizu senzoru od 0-3, gdje je 3 najbliže. Nadalje, u ovisnosti o polju stanja, iscrtavaju se pravokutnici.

```
void turnOnSensors();
void draw();
int state[6] = { 0 };
float distances[6] = { 0 };
bool played;
POINT p;
```

2. Klasa Sensor

Kako ne bi morao ručno kodirati koordinate senzora, napravio sam klasu Sensor koja enkapsulira dva podatka koji predstavljaju koordinate senzora. Kako imam 6 senzora, napravio sam 6 instanci te klase (vidi funkciju turnOnSensors()).

```
class Sensor {

public:
	int x;
	int y;

	Sensor(int xpos, int ypos) {
		x = xpos;
		y = ypos;}

	int getX() { return x; }
	int getY() { return y; }

};

//Front Sensors Initialization
Sensor frontLeftSensor(680, 647);
Sensor frontMidSensor(660, 570);
Sensor frontRightSensor(680, 493);

//Rear Sensors Initialization
Sensor rearRightSensor(1135, 493);
Sensor rearMidSensor(1155, 570);
Sensor rearLeftSensor(1135, 647);
```

3.Funkcija distance()

Funkcija distance() je pomoćna funkcija koja prima koordinate točke i jedan Sensor. Ono sto vraća jest udaljenost točke od Senzora

```
float distance(int x1, int y1,
Sensor Sensor)
{
	int x2, y2;
	x2 = Sensor.x;
	y2 = Sensor.y;
	return sqrt(pow(x2 - x1,
2) + pow(y2 - y1, 2) * 1.0);
		}
```

4. Funkcija glDrawSensors()

Ova funkcija je također pomoćna funkcija, služi za iscrtavanje plavih kvadratića koji predstavljaju senzore.

```
void glDrawSensors() {

	//drawing sensor 0
	glColor3f(0.0, 0.0, 1.0);
	glVertex3f(-1.31f, -0.41f, 0.0f); // XYZ left, top
	glVertex3f(-1.27f, -0.41f, 0.0f); // XYZ right, top
	glVertex3f(-1.27f, -0.45f, 0.0f); // XYZ right, bottom
	glVertex3f(-1.31f, -0.45f, 0.0f); // XYZ left, bottom

	//drawing sensor 1
	glVertex3f(-1.42f, 0.02f, 0.0f); // XYZ left, top
	glVertex3f(-1.38f, 0.02f, 0.0f); // XYZ right, top
	glVertex3f(-1.38f, -0.02f, 0.0f); // XYZ right, bottom
	glVertex3f(-1.42f, -0.02f, 0.0f); // XYZ left, bottom

	//drawing sensor 2
	glVertex3f(-1.31f, 0.41f, 0.0f); // XYZ left, top
	glVertex3f(-1.27f, 0.41f, 0.0f); // XYZ right, top
	glVertex3f(-1.27f, 0.45f, 0.0f); // XYZ right, bottom
	glVertex3f(-1.31f, 0.45f, 0.0f); // XYZ left, bottom

	//drawing sensor 3
	glVertex3f(1.27f, 0.41f, 0.0f); // XYZ left, top
	glVertex3f(1.23f, 0.41f, 0.0f); // XYZ right, top
	glVertex3f(1.23f, 0.45f, 0.0f); // XYZ right, bottom
	glVertex3f(1.27f, 0.45f, 0.0f); // XYZ left, bottom


	//drawing sens 4
	glVertex3f(1.37f, 0.02f, 0.0f); // XYZ left, top
	glVertex3f(1.33f, 0.02f, 0.0f); // XYZ right, top
	glVertex3f(1.33f, -0.02f, 0.0f); // XYZ right, bottom
	glVertex3f(1.37f, -0.02f, 0.0f); // XYZ left, bottom

	//drawing sensor 5
	glVertex3f(1.27f, -0.41f, 0.0f); // XYZ left, top
	glVertex3f(1.23f, -0.41f, 0.0f); // XYZ right, top
	glVertex3f(1.23f, -0.45f, 0.0f); // XYZ right, bottom
	glVertex3f(1.27f, -0.45f, 0.0f); // XYZ left, bottom

	return;
};
```

5. Funkcija turnOnSensors()

Ova funkcija sadrži logiku za paljenje pojedinih senzora. U polje sprema udaljenosti između položaja miša i senzora, te uz pomoć tih podataka polje stanja senzora postavlja na 1,2 ili 3 ovisno o tome koliko je velika udaljenost. Nakon svih provjera i izračuna, poziva se funkcija draw().

```
void turnOnSensors() {

	//Calculate Cursor Distance to Each Sensor
	float distances[6] = { 0 };
	distances[0] = distance(p.x, p.y, frontLeftSensor);
	distances[1] = distance(p.x, p.y, frontMidSensor);
	distances[2] = distance(p.x, p.y, frontRightSensor);
	distances[3] = distance(p.x, p.y, rearRightSensor);
	distances[4] = distance(p.x, p.y, rearMidSensor);
	distances[5] = distance(p.x, p.y, rearLeftSensor);

	//frontLeftSenSor Managment
	if (distances[0] <= 85) {
		state[0] = 1;
		if (distances[0] <= 65) {
			state[0] = 2;
			if (distances[0] <= 45) {
				state[0] = 3;
			}
		}
	}
	else { state[0] = 0; }

	//…isti kod za svaki od senzora


	draw();
	return;
}
```

6.Funckija draw()

Kako samo ime kaže, ova funkcija crta pravokutnike. Dodatna funkcionalnost je da i pušta zvukove, ovisno o vrijednosti u polju stanja senzora. Sastoji se od 6 Switcheva, po jednog za svaki senzor. Ono što ona zapravo radi jest provjera koliko kvadrata treba isrctati, iscrtava ih te pušta zvukove.Konkretno, na primjeru prednjeg lijevog senzora, ako je stanje 3, prepreka je na najbliža senzoru te će crtati 3 pravokutnika i puštati zvuk left_3.wav. U slučaju da je stanje 2, crta 2 pravokutnika i pušta left_2.wav. Analogno, za stanje 1 i ostale senzore.

```
void draw() {
	glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT);
	glBegin(GL_QUADS);


	// drawing a background white rectangle
	glColor3f(1, 1, 1); // choosing a color
	glTexCoord2f(0, 1); glVertex3f(-2, -1, 0);
	glTexCoord2f(1, 1); glVertex3f(2, -1, 0);
	glTexCoord2f(1, 0); glVertex3f(2, 1, 0);
	glTexCoord2f(0, 0); glVertex3f(-2, 1, 0);

	glDrawSensors();


	switch (state[0]) {
		// Front Left
	case 3:
		// drawing a rectangle
		glColor3f(1.0, 0.0, 0.0);
		glVertex3f(-1.57f, -0.45f, 0.0f);
		glVertex3f(-1.50f, -0.41f, 0.0f);
		glVertex3f(-1.38f, -0.61f, 0.0f);
		glVertex3f(-1.44f, -0.65f, 0.0f);

		// drawing a rectangle
		glColor3f(1.0, 1.0, 0.0);
		glVertex3f(-1.70f, -0.45f, 0.00f);
		glVertex3f(-1.63f, -0.41f, 0.0f);
		glVertex3f(-1.44f, -0.71f, 0.00f);
		glVertex3f(-1.50f, -0.75f, 0.0f);

		// drawing a rectangle
		glColor3f(0.0, 1.0, 0.0);
		glVertex3f(-1.83f, -0.46f, 0.0f);
		glVertex3f(-1.75f, -0.41f, 0.0f);
		glVertex3f(-1.49f, -0.80f, 0.0f);
		glVertex3f(-1.57f, -0.85f, 0.0f);

		if(state[0]>=state[1])
			PlaySound(TEXT("D:\\left_3.wav"),
NULL, SND_FILENAME | SND_LOOP);
```

```
case 2:
	// drawing a rectangle
	glColor3f(1.0, 1.0, 0.0);
	glVertex3f(-1.70f, -0.45f, 0.00f);
	glVertex3f(-1.63f, -0.41f, 0.0f);
	glVertex3f(-1.44f, -0.71f, 0.00f);
	glVertex3f(-1.50f, -0.75f, 0.0f);
	// drawing a rectangle
	glColor3f(0.0, 1.0, 0.0);
	glVertex3f(-1.83f, -0.46f, 0.0f);
	glVertex3f(-1.75f, -0.41f, 0.0f);
	glVertex3f(-1.49f, -0.80f, 0.0f);
	glVertex3f(-1.57f, -0.85f, 0.0f);

if(state[0]>=state[1])
	PlaySound(TEXT("D:\\left_2.wav"),
NULL, SND_FILENAME | SND_LOOP);
		break;

case 1:
	// drawing a rectangle
	glColor3f(0.0, 1.0, 0.0);
	glVertex3f(-1.83f, -0.46f, 0.0f);
	glVertex3f(-1.75f, -0.41f, 0.0f);
	glVertex3f(-1.49f, -0.80f, 0.0f);
	glVertex3f(-1.57f, -0.85f, 0.0f);


if (state[0] >= state[1])
	PlaySound(TEXT("D:\\left_1.wav"),
NULL, SND_FILENAME | SND_LOOP);
		break;
	}
```